Booth multiplier for trigonometric functions

Publication number: EP0718746
Publication date: 1996-06-26
Inventor: MARTINEZ GEORGES (FR); CHAUVE NICOLAS (FR)
Applicant: PHILIPS ELECTRONIQUE LAB (FR); PHILIPS ELECTRONICS NV (NL)
Classification:
- international: ***G06F7/52; G06F1/035; G06F7/523; G06F7/533; G06F7/548; G06F17/14;*** **G06F1/02; G06F7/48; G06F17/14;** (IPC1-7): G06F1/035; G06F7/52; G06F7/544
- European: G06F1/035B
Application number: EP19950203453 19951212
Priority number(s): FR19940015410 19941221

Also published as:
US5684730 (A1)
JP8241187 (A)
EP0718746 (B1)
DE69534097T (T2)

Cited documents:
US5276633
US4159527
FR2208554
GB2152715

Report a data error here

Abstract of **EP0718746**

The multiplicand (X) is multiplied (10) by a factor (Y) taken from a trigonometric function store (14) and encoded (12). The argument (theta) of the function is entered a binary word in the store, which holds only values relative to a single quadrant. If the input value lies outside that quadrant, a conversion unit (11) transforms it in accordance with the logic states of the most significant bits (BP-1,BP-2). The values of the trigonometric function may be obtd. by interpolation.

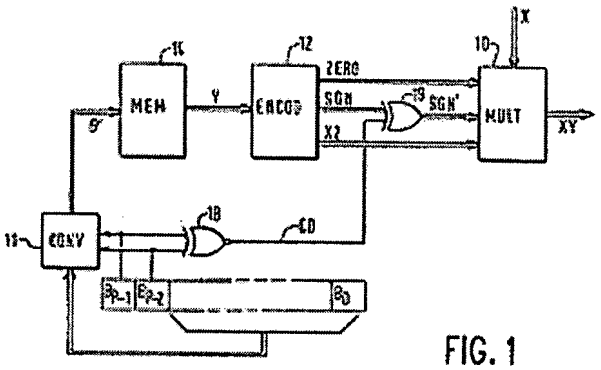

FIG. 1

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 718 746 A1

(12) DEMANDE DE BREVET EUROPEEN

(43) Date de publication:
26.06.1996 Bulletin 1996/26

(51) Int Cl.[6]: **G06F 1/035**, G06F 7/52, G06F 7/544

(21) Numéro de dépôt: **95203453.6**

(22) Date de dépôt: **12.12.1995**

(84) Etats contractants désignés:
DE FR GB

(30) Priorité: **21.12.1994 FR 9415410**

(71) Demandeurs:
- **LABORATOIRES D'ELECTRONIQUE PHILIPS S.A.S.**
  94450 Limeil-Brévannes (FR)
  Etats contractants désignés:
  FR
- **PHILIPS ELECTRONICS N.V.**
  5621 BA Eindhoven (NL)
  Etats contractants désignés:
  DE GB

(72) Inventeurs:
- **Martinez, Georges, c/o Société Civile S.P.I.D.**
  75008 Paris (FR)
- **Chauve, Nicolas, c/o Société Civile S.P.I.D.**
  75008 Paris (FR)

(74) Mandataire: **Chaffraix, Jean**
Société Civile S.P.I.D.
156, Boulevard Haussmann
75008 Paris (FR)

(54) Multiplieur de booth pour fonctions trigonométriques

(57) Circuit pour multiplier des données conformément à un algorithme de Booth (10) dans lequel le codage (12) des commandes est adapté aux caractéristiques de symétrie d'une fonction trigonométrique. On stocke les valeurs de la fonction relatives à un seul quadrant dans une mémoire (14) et une unité de conversion (11) permet d'utiliser le circuit quel que soit le quadrant. Les valeurs de la fonction trigonométrique peuvent être obtenues par interpolation. Application aux calculs de transformées de Fourier ou de transformées DCT.

Application : transformées de Fourier, transformée DCT.

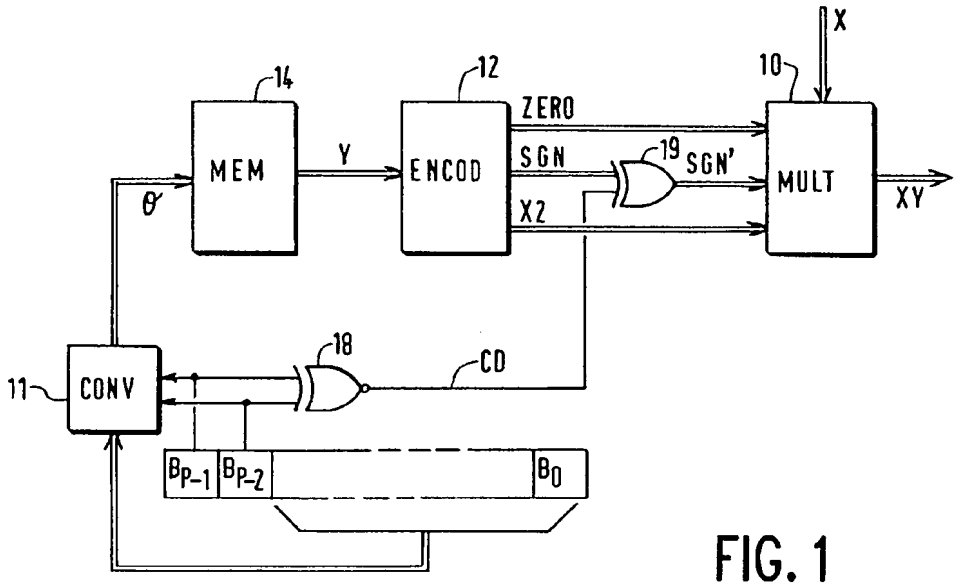

FIG. 1

EP 0 718 746 A1

Description

L'invention concerne un circuit pour multiplier des données par des valeurs d'une fonction trigonométrique d'une variable d'entrée, le circuit comprenant :

- des premiers moyens pour exécuter une multiplication en appliquant un algorithme de Booth,
- des seconds moyens pour générer des commandes appliquées aux premiers moyens, dont des commandes de signe, conformément à l'algorithme de Booth,
- des troisièmes moyens pour délivrer les valeurs de la fonction trigonométrique.

Elle concerne aussi un système de traitement numérique utilisant un tel circuit et plus particulièrement un système pour un calcul de transformées de Fourier ou autres. Elle concerne aussi un système de transmission numérique comprenant un circuit numérique de récupération de porteuse.

Il est courant d'utiliser des circuits pour le calcul de transformées de Fourier dans le traitement numérique de signal. Une transformée de Fourier directe et une transformée de Fourier réciproque concernent pour la première une transformation de signaux représentés dans le domaine temporel en signaux représentés dans le domaine fréquenciel et pour la seconde une transformation réciproque. Ces transformations qui mettent en oeuvre des fonctions exponentielles complexes impliquent la manipulation de fonctions trigonométriques. Ces manipulations nécessitent de réaliser un nombre important de multiplications par des fonctions sinus/cosinus.

Il est donc souhaitable d'utiliser des techniques rapides de multiplication pour diminuer les temps de calcul en y associant, si possible, l'utilisation d'un matériel réduit pour limiter les coûts de mise en oeuvre.

Une technique connue répondant à ces spécifications est celle de l'algorithme de Booth qui consiste à grouper les bits des opérandes d'une multiplication par groupes de 3 bits et à générer un nombre réduit de signaux de commandes, par exemple trois commandes seulement, par groupe de trois bits :

- ZERO : ne rien faire
- SGN commande le signe de l'opération à effectuer
- X2 : détermine si le multiplicande doit être ajouté/retranché (selon SGN) une fois ou 2 fois aux sommes partielles.

Un multiplieur qui met en oeuvre l'algorithme de Booth, à l'aide de ces trois commandes, réalise des multiplications rapides.

Néanmoins, des techniques permettant d'améliorer encore ces performances sont toujours recherchées.

On connaît le document intitulé : "Parallel architecture modified Booth multiplier" de A.R. COOPER, IEE Proceedings Vol. 135, Pt. G., N° 3, June 1988, qui décrit un multiplieur de Booth amélioré.

En regroupant les opérations sélectionnées par les sorties des encodeurs de Booth et en les faisant opérer en parallèle, il est possible de simplifier le déroulement des opérations réalisées selon l'algorithme de Booth. Un tel multiplieur peut être utilisé dans tout calcul et ne tient pas compte de la particularité de l'application donc des données qui sont à multiplier entre-elles.

Un objet de l'invention est de simplifier encore la construction d'un multiplieur de Booth en tenant compte de la spécificité des données notamment dans l'exploitation des données trigonométriques.

Ce but est atteint avec un circuit pour multiplier des données par des valeurs d'une fonction trigonométrique d'une variable d'entrée, tel qu'il est décrit dans le préambule, dans lequel les troisièmes moyens délivrent des valeurs de la fonction trigonométrique sur un domaine limité de la variable d'entrée, et en ce que le circuit comporte des moyens pour modifier les commandes de signe d'après des états logiques de bits de poids fort de la variable d'entrée et des moyens de conversion de la variable d'entrée pour faire que le circuit fonctionne avec toute valeur de la variable d'entrée.

Ainsi avantageusement, en stockant uniquement les valeurs pour un domaine limité de la variable d'entrée (par exemple $-\pi/2 \leq \theta \leq 0$ pour une fonction cos θ), on réduit très sérieusement la taille de la mémoire. De plus, en faisant que la commande de signe soit modifiée après les moyens qui génèrent classiquement les commandes, on réalise simplement avec peu de matériel la commande des inversions de signe qu'il est nécessaire de faire pour réaliser la multiplication même lorsque la valeur θ d'entrée n'est pas dans le quadrant trigonométrique correspondant aux valeurs stockées. En particulier, ceci évite d'avoir à ajouter un multiplieur par - 1 pour effectuer des changements de quadrants ce qui consomme à la fois du temps de calcul et de la surface de circuit intégré.

Préférentiellement les troisièmes moyens délivrent des valeurs négatives exprimées en complément à 2. Ainsi avantageusement, pour un format binaire donné (nombre de bits et répartition des poids binaires), il est possible de représenter exactement la valeur - 1, ce qui ne serait pas possible avec la valeur + 1, exprimée en complément à deux, sans introduire une imprécision de calcul de 1 bit du poids le plus faible. Ceci constitue un avantage important dans le cas de multiplications effectuées à l'aide de l'algorithme de Booth décrit.

Selon un mode particulier de réalisation, les moyens qui délivrent les valeurs de la fonction trigonométrique calculent ces valeurs par interpolation autour d'un nombre limité de valeurs de fonction préalablement mémorisées. Ceci réduit également sérieusement la taille de la mémoire.

Ces différents aspects de l'invention et d'autres encore seront apparents et élucidés à partir des modes de réalisation décrits ci-après.

L'invention sera mieux comprise à l'aide des figures suivantes données à titre d'exemples non limitatifs qui représentent :

Figure 1 : un schéma d'un circuit multiplieur selon l'invention.

Figure 2 : un schéma d'un circuit multiplieur selon l'invention muni de moyens pour interpoler des valeurs de fonction entre des valeurs stockées dans un premier bloc-mémoire.

Figure 3 : un graphique d'une partie agrandie d'une fonction $F(\theta) = \cos(\theta)$.

Figure 4 : un schéma d'un circuit d'interpolation.

La figure 1 représente un cicuit multiplieur selon l'invention comprenant des moyens 10 MULT pour multiplier un multiplicande X par un multiplicateur Y conformément à l'algorithme de Booth, des moyens 12 ENCOD pour générer des commandes conformément à cet algorithme de Booth afin de permettre aux moyens 10 d'opérer, et des moyens 14 MEM pour délivrer les valeurs du multiplicateur Y au multiplieur 10.

L'algorithme de Booth consiste à ajouter un zéro en bit de poids faible puis à regrouper les bits du multiplicateur, exprimé en complément à 2, en groupes de trois bits consécutifs à partir du bit de poids faible ajouté et à en déduire les opérations à effectuer. Deux groupes consécutifs se chevauchent par un bit. Les trois bits d'un groupe imposent le type d'opérations à effectuer :

Code	Opération
(001; 010)	- additionner le multiplicande à la somme partielle
(000; 111)	- ne rien faire
(011)	- additionner deux fois le multiplicande à la somme partielle
(101, 110)	- soustraire le multiplicande de la somme partielle
(100)	- soustraire 2 fois le multiplicande de la somme partielle.

Ces cinq commandes peuvent être ramenées à trois commandes seulement :

- ZERO - ne rien faire
- SGN - commande le signe de l'opération à effectuer
- X 2 - détermine si le multiplicande doit être ajouté/retranché (selon SGN) une fois ou 2 fois de la somme partielle.

Il y a donc autant de fois les trois commandes précécentes qu'il y a de triplets de bits dans le multiplicateur Y. Si celui-ci contient b bits, le nombre de triplets est égal à l'entier le plus proche soit de b/2 soit de b/2 + 1 selon la parité de b.

L'invention concerne des modifications de la commande du signe SGN délivrée normalement par un encodeur de Booth (12).

Le multiplieur 10 de Booth reçoit le multiplicande X et plusieurs lots des trois commandes précédentes. Le multiplicateur Y est codé par les moyens de codage 12 ENCOD pour délivrer la pluralité de commandes ZERO, SGN et X2. Dans l'invention, le multiplicateur Y représente des valeurs d'une fonction trigonométrique, par exemple cos θ. Il est obtenu en adressant des moyens de mémorisation 14, par exemple une mémoire MEM, qui ont préalablement été chargés par des valeurs de la fonction trigonométrique. L'argument θ de la fonction trigonométrique est fourni en entrée du circuit selon un mot binaire $B_{P-1}....B_0$. En utilisant les propriétés de symétrie de la fonction cos θ, il est possible de ne stocker dans la mémoire 14 que les valeurs cos θ correspondant à $\pi/2 \leq \theta \leq \pi$, c'est-à-dire des valeurs négatives de cos θ. Les deux bits de poids fort B_{P-1} et B_{P-2} permettent de déduire les valeurs de cos θ pour les autres valeurs de θ.

Selon l'invention, on contrôle la commande de signe SGN de l'encodeur de Booth. Une porte 18 reçoit B_{P-1} et B_{P-2} et délivre un signal CD. Une famille 19 de portes OU - exclusif reçoit le signal CD et les signaux SGN issus de chaque triplet de bits de Y. En sortie, on obtient de nouveaux signaux de signe SGN' qui sont utilisés pour sélectionner le type d'opération à effectuer dans le multiplieur de Booth 10. Le mot binaire $B_{P-1}... B_0$ est codé en complément à 2. Chaque nouvelle commande de signe SGN' est alors déduite de la table I de vérité suivante :

TABLE I

B_{P-1}	B_{P-2}	CD	SGN	SGN'
0	1	0	0	0
0	0	1	0	1

TABLE I (suite)

1	0	0	1	1
1	1	1	1	0

Préférentiellement, la mémoire 14 stocke des valeurs négatives de cos θ donc avec $\pi/2 < \theta < \pi$ ou $\pi < \theta < 3\pi/2$.

Lorsque la valeur θ qui arrive en entrée du circuit n'est pas dans le quadrant correspondant aux valeurs stockées dans la mémoire, alors une unité de conversion 11 transforme la valeur θ d'entrée d'après les états logiques des bits de poids élevé B_{P-1}, B_{P-2}. Selon le contenu de la mémoire 14 (sinus ou cosinus), la conversion va consister à convertir θ en $\pi/2 - \theta$ et à modifier éventuellement le signe selon le quadrant concerné.

Pour réduire encore les moyens matériels, les moyens 14 qui délivrent la fonction trigonométrique peuvent opérer par interpolation autour d'un nombre limité de valeurs de la fonction trigonométrique préalablement memorisées (figure 2).

Le mot binaire $B_{P-1}... B_0$ est alors séparé en trois champs :

- un premier champ I formé par des bits de poids faible du mot binaire,
- un second champ II formé par des bits médians du mot binaire,
- un troisième champ III, formé par les bits $B_{P-1}... B_{P-2}$, qui sert à contrôler les commandes de signe comme exposé auparavant.

Les premier et second champs I et II servent à adresser (figure 2) une unité de conversion 11 CONV qui transforme les valeurs d'entrée θ en valeur de θ située dans le quadrant pour lequel on a mémorisé préalablement un nombre limité de valeurs de la fonction trigonométrique dans un premier bloc-mémoire 142. Les valeurs stockées peuvent être préalablement corrigées par des coefficients correctifs spécifiques destinés à réduire des erreurs engendrées par la méthode d'interpolation utilisée.

L'unité de conversion adresse aussi un second bloc-mémoire 144 dans lequel on a mémorisé préalablement des coefficients d'interpolation calculés pour les valeurs de la fonction trigonométrique qui ont été stockées dans le premier bloc-mémoire. Les coefficients d'interpolation peuvent être une dérivée vraie ou une dérivée approximative formée par exemple par la pente moyenne du segment joignant deux valeurs stockées consécutives de la fonction.

Ainsi en adressant les deux blocs-mémoires avec le champ médian II (avec ou sans conversion selon le cas), par exemple on peut lire la valeur de la fonction cos θ (avec corrections) et celle des coefficients d'interpolation pour cette valeur θ.

Le champ I de bits de poids faibles est alors exploité pour calculer une quantité interpolée liée à l'incrément fourni par le champ I.

La figure 3 représente une partie d'une fonction $F(\theta) = \cos(\theta)$ connue par un nombre limité de valeurs $F(\theta_{nk}) = \cos(\theta_{nk})$ [point A], $F(\theta_{(n+1)k}) = \cos(\theta_{(n+1)k})$ [point B], pour des valeurs θ_{nk}, $\theta_{(n+1)k}$ de la variable θ. Sur le graphique n est l'indice courant des points A, B..., avec $0 \leq n \leq \frac{N}{k}-1$, k étant un nombre entier déterminant le facteur d'interpolation, avec $\theta_n = 2\pi\frac{n}{N}$.

Lorsqu'une valeur quelconque θ_{nk+i} de la variable est présentée en entrée du circuit, celui-ci calcule, par interpolation, une valeur interpolée proche de la valeur $F(\theta n_{k+i})$ [point C]. Sous cette forme, l'indice i est un indice courant variant de 0 à k-1 afin de différencier k valeurs intermédiaires entre θ_{nk} et $\theta_{(n+1)k}$.

Pour représenter N valeurs de la variable θ, il faut $\log_2 N$ bits. Une valeur quelconque θ_{nk+i} de la variable est représentée par un mot binaire pour lequel on ignore les deux bits de poids forts du début pour ne considérer que les valeurs appartenant à un seul quadrant. Le mot binaire 13 (figure 2) comprend un premier champ 13A comportant des bits de poids fort et un second champ 13B comportant des bits de poids faible.

Selon l'invention, l'interpolation est effectuée en exploitant un nombre N/k de valeurs connues de la fonction. Ces N/k valeurs connues sont préférentiellement stockées préalablement dans un bloc-mémoire 142. Les valeurs connues $F(\theta_{nk})$ sont celles qui correspondent aux $\log_2 N/k$ bits de poids fort de la variable d'entrée.

Pour les N/k valeurs connues de la fonction (points A, B, etc), selon l'invention, on stocke également, préférentiellement dans des moyens de mémorisation 144, N/k coefficients d'interpolation $C(\theta_{nk})F$ Ceux-ci peuvent être des valeurs de dérivée $F'(\theta_{nk})$ de la fonction $F(\theta_{nk})$. Ces valeurs de dérivée peuvent être des valeurs moyennes de dérivée calculées autour de chaque point connu A, B, etc. Préférentiellement, ces valeurs de coefficient d'interpolation sont égales à la pente des segments reliant deux points connus consécutifs.

Ces valeurs de coefficient d'interpolation peuvent également être formées par des valeurs vraies de dérivée calculées aux points connus A, B, etc.

Pour limiter les imprécisions, on stocke dans le bloc-mémoire 142 des valeurs connues de la fonction préalablement corrigées par des termes correctifs. Le bloc-mémoire stocke donc des valeurs $F(\theta_{nk}) + cor(\theta_{nk})$. Le terme cor (θ_{nk}) est préalablement calculé pour chaque point connu pour minimiser l'erreur quadratique moyenne entre les valeurs

interpolées et la fonction trigonométrique parfaite.

Lorsqu'une valeur d'entrée θ_{nk+i} est présentée à l'entrée du circuit, le champ 13A de bits de poids fort adresse (connexion 15) le bloc-mémoire 142 et les moyens de mémorisation 144 :

- le bloc-mémoire délivre $F(\theta_{nk}) + cor(\theta_{nk})$ constituant la valeur approchée de la valeur interpolée à déterminer (connexion 17),
- les moyens de mémorisation délivrent les coefficients d'interpolation $C'(\theta_{nk})$ (connexion 16).

Considérons le cas où la fonction $F(\theta_{nk})$ est une fonction cosinus et où les valeurs de coefficients d'interpolation sont déduites de la pente d'un segment y = a x+b. Ce segment passe au voisinage des deux points connus A et B pour indiquer que des coefficients de corrections interviennent dans le calcul des valeurs interpolées.

Une valeur approchée y de cos (θ_{nk+i}) [point D] est alors telle que :

$$y = \frac{\cos \theta_{nk+k} - \cos \theta_{nk}}{\theta_{(nk+k)} - \theta_{nk}} (\theta_{nk+i} - \theta_{nk}) + \cos \theta_{nk}$$

avec

$$\theta_{n_{k+i}} = \frac{2\pi}{N} (nk + i)$$

$$0 \leq i \leq k\text{-}1 \qquad 0 \leq n \leq \frac{N}{k} - 1$$

Après des manipulations trigonométriques connues, on obtient :

$$Y = \left[-2 \sin \frac{(2n+1)k\pi}{N} . \sin \frac{k\pi}{N}\right] \frac{i}{k} + \cos \frac{2\pi}{N} nk$$

Pour obtenir une valeur interpolée *côs* (θ_{nk+i}) plus proche de la valeur exacte cos (θ_{nk+i}), on introduit un terme de correction cor (θ_{nk}). Cette valeur interpolée est située entre les ordonnées des points D et C. Chaque terme de correction cor (θ_{nk}) est déterminé en rendant minimum la somme $\Sigma(côs\ \theta_{nk+i} - \cos \theta_{nk+i})^2$ pour $0 \leq i \leq k\text{-}1$.

La valeur approchée côs (θ_{nk+i}) s'écrit alors :

$$côs\ (\theta_{nk+i}) = \underbrace{\left\{-\frac{2}{k} \sin \left(\frac{2n+1}{N} . k\pi\right) . \sin \left(\frac{k\pi}{N}\right)\right\}}_{F'} i + \cos \left(\frac{2\pi}{N} nk\right) + cor\ (\theta_{nk})$$

Le circuit comprend des moyens de calcul 145, par exemple un multiplieur/accumulateur qui reçoit $C'(\theta_{nk})$ et le champ 13B formé des bits de poids faible de la valeur d'entrée θ_{nk+i}.

Dans une première étape, les moyens de calcul 145 calculent le résultat partiel :

$$C(\theta_{nk}).i$$

et dans une seconde étape, ils ajoutent ce résultat partiel à la valeur approchée $F(\theta_{nk}) + cor(\theta_{nk})$ pour délivrer la valeur interpolée : $\hat{F}(\theta_{nk+i})$ soit :

$$\hat{F}(\theta_{nk+i}) = [(C'(\theta_{nk}) . i] + F(\theta_{nk}) + cor(\theta_{nk})$$

Ainsi, il est possible de réduire la taille de la seconde mémoire en ne stockant qu'un nombre limité de valeurs sur un nombre réduit de bits tout en obtenant une précision élevée sur le résultat, ce qui est particulièrement intéressant dans la réalisation de circuits intégrés.

Dans le cas où ils délivrent des valeurs interpolées, les troisièmes moyens comprennent :

- le premier bloc-mémoire 142 pour mémoriser un nombre limité de valeurs négatives de la fonction correspondant au second champ de bits, à des adresses issues du second champ de bits,
- le second bloc-mémoire 144 pour mémoriser des coefficients d'interpolation associés aux valeurs mémorisées dans le premier bloc-mémoire, à des adresses issues du second champ de bits,
- des moyens d'interpolation

  - pour lire, dans le premier bloc-mémoire, une valeur approchée de la fonction en l'adressant par le second champ de bits de la variable d'entrée,
  - pour lire, dans le second bloc-mémoire, le coefficient d'interpolation en l'adressant par le second champ de bits de la variable d'entrée,
  - pour calculer une quantité interpolée en multipliant le premier champ de bits de la variable d'entrée par le

coefficient d'interpolation adressé, la quantité interpolée étant totalisée avec la valeur approchée pour donner la valeur interpolée.

Le circuit peut être utilisé dans un système de traitement numérique de données pour effectuer des calculs de transformées de Fourier, de transformées en cosinus discrètes, ou d'autres transformées faisant appel à des fonctions trigonométriques.

Le circuit peut aussi être utilisé dans un système de transmission numérique dans lequel le circuit selon l'invention entre dans la réalisation d'un circuit numérique de récupération de porteuse. Dans ce cas, la variable d'entrée est une erreur de phase θ à partir de laquelle le circuit multiplie des valeurs cos θ et sin θ par un multiplicande formé par un symbole qui doit subir une correction de phase pour synchroniser la récupération de porteuse.

Revendications

1. Circuit pour multiplier des données par des valeurs d'une fonction trigonométrique d'une variable d'entrée, le circuit comprenant :

   - des premiers moyens pour exécuter une multiplication en appliquant un algorithme de Booth,
   - des seconds moyens pour générer des commandes appliquées aux premiers moyens, dont des commandes de signe, conformément à l'algorithme de Booth,
   - des troisièmes moyens pour délivrer les valeurs de la fonction trigonométrique,

   caractérisé en ce que les troisièmes moyens délivrent des valeurs de la fonction trigonométrique sur un domaine limité de la variable d'entrée, et en ce que le circuit comporte des moyens pour modifier les commandes de signe d'après des états logiques de bits de poids fort de la variable d'entrée et des moyens de conversion de la variable d'entrée pour faire que le circuit fonctionne avec toute valeur de la variable d'entrée.

2. Circuit selon la revendication 1 caractérisé en ce que les troisièmes moyens délivrent des valeurs négatives de la fonction trigonométrique.

3. Circuit selon les revendications 1 ou 2 caractérisé en ce que les commandes de signe sont inversées lorsque les deux bits de poids fort de la variable d'entrée sont identiques.

4. Circuit selon la revendication 3 caractérisé en ce que les troisièmes moyens calculent la valeur de la fonction trigonométrique par interpolation autour d'un nombre limité de valeurs de fonction préalablement mémorisées.

5. Circuit selon la revendication 4 caractérisé en ce que la variable d'entrée étant représentée par un premier champ de bits de poids faible, par un second champ de bits de poids intermédiaire, et par un troisième champ de bits de poids fort, les troisièmes moyens calculent, par interpolation, une valeur interpolée de la fonction trigonométrique pour la valeur de la variable d'entrée, les troisièmes moyens comprenant :

   - un premier bloc-mémoire pour mémoriser un nombre limité de valeurs négatives de la fonction correspondant au second champ de bits, à des adresses issues du second champ de bits,
   - un second bloc-mémoire pour mémoriser des coefficients d'interpolation associés aux valeurs mémorisées dans le premier bloc-mémoire, à des adresses issues du second champ de bits,
   - des moyens d'interpolation

     - pour lire, dans le premier bloc-mémoire, une valeur approchée de la fonction en l'adressant par le second champ de bits de la variable d'entrée,
     - pour lire, dans le second bloc-mémoire, le coefficient d'interpolation en l'adressant par le second champ de bits de la variable d'entrée,
     - pour calculer une quantité interpolée en multipliant le premier champ de bits de la variable d'entrée par le coefficient d'interpolation adressé, la quantité interpolée étant totalisée avec la valeur approchée pour donner la valeur interpolée.

6. Système de traitement numérique de données caractérisé en ce qu'il comprend au moins un circuit selon une des revendications 1 à 5 pour un calcul de transformée de Fourier ou de transformée en cosinus discrète ou de transformée faisant appel à une fonction trigonométrique.

7. Système de transmission numérique caractérisé en ce qu'il comprend au moins un circuit selon une des revendications 1 à 5 pour réaliser un circuit numérique de récupération de porteuse.

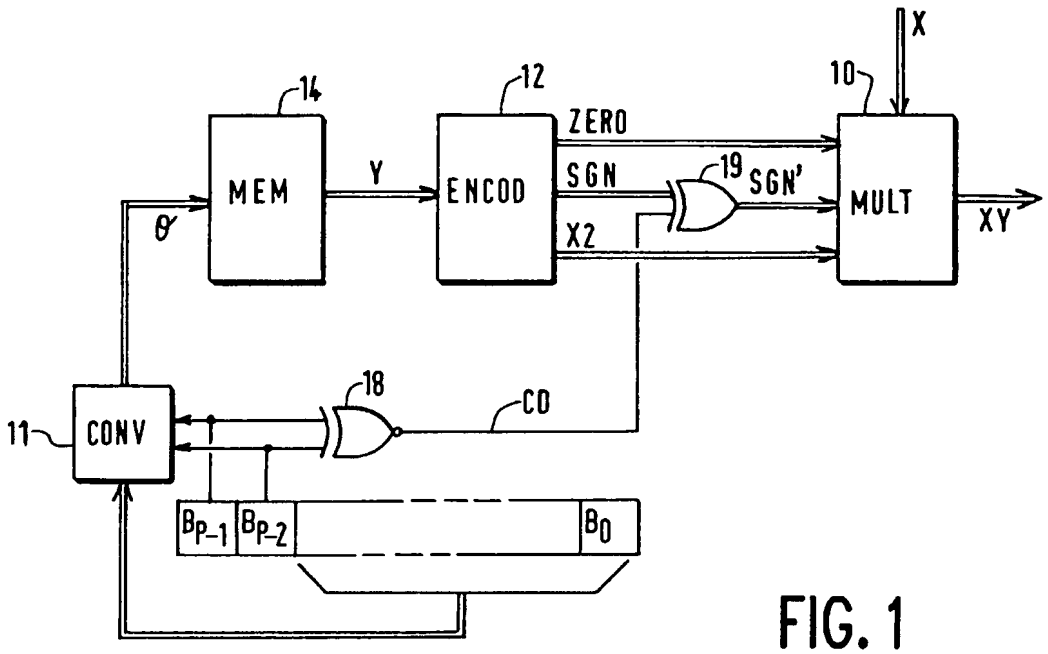
X
10
14
12
ZERO
MEM
Y
ENCOD
SGN
19
SGN'
MULT
XY
θ
X2
18
CO
11
CONV
B_{P-1}
B_{P-2}
B_0

FIG. 1

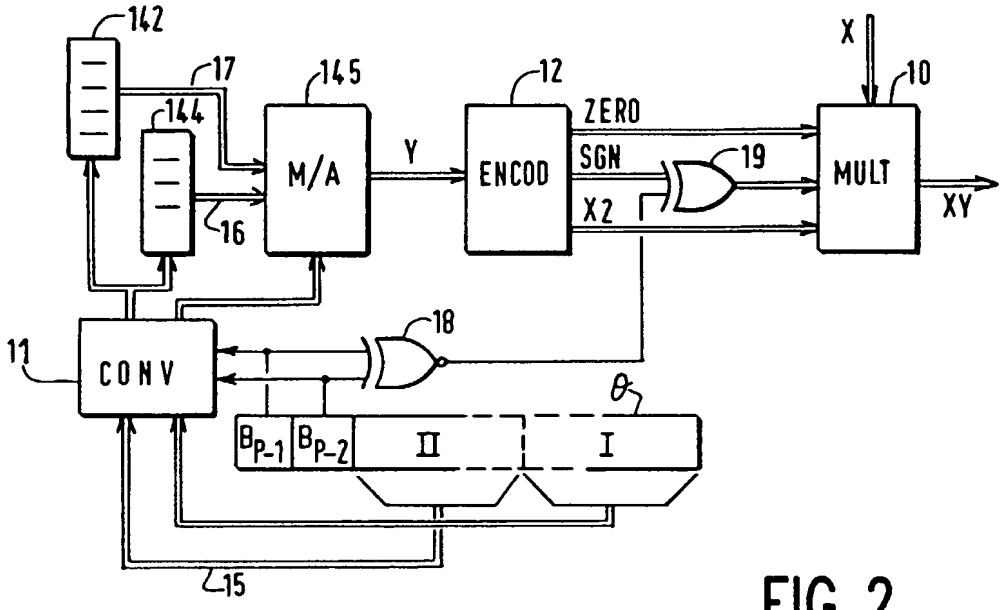
142
17
145
12
X
10
144
M/A
Y
ENCOD
ZERO
SGN
19
MULT
XY
16
X2
18
11
CONV
θ
B_{P-1}
B_{P-2}
II
I
15

FIG. 2

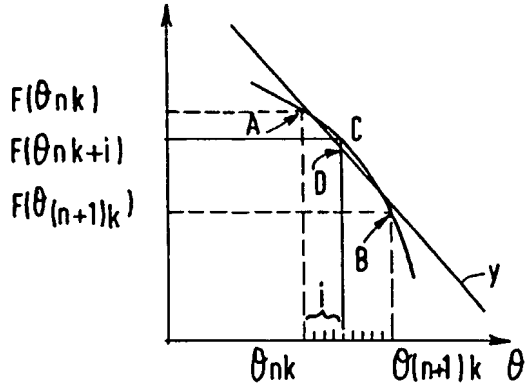
F(θnk)
F(θnk+i)
F(θ(n+1)k)
A
C
D
B
i
y
θnk
θ(n+1)k
θ

FIG. 3

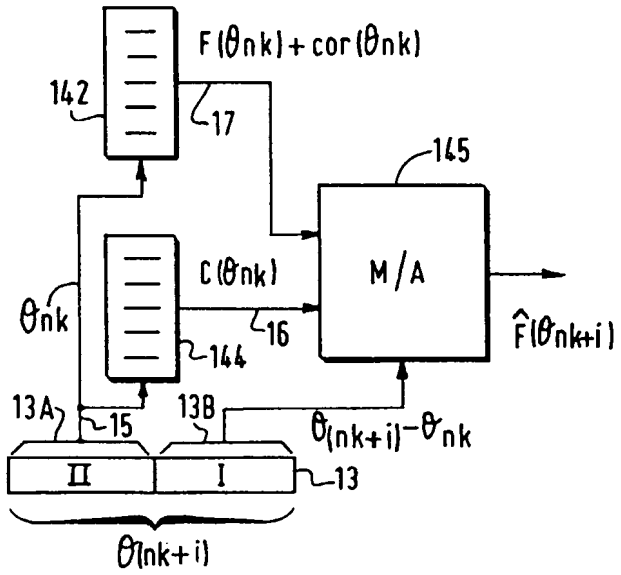
F(θnk)+cor(θnk)
142
17
145
C(θnk)
M/A
θnk
16
144
F̂(θnk+i)
13A
15
13B
θ(nk+i)−θnk
II
I
13
θ(nk+i)

FIG. 4

Office européen des brevets

RAPPORT DE RECHERCHE EUROPEENNE

Numero de la demande
EP 95 20 3453

DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendication concernée	CLASSEMENT DE LA DEMANDE (Int.Cl.6)
X	US-A-5 276 633 (FOX JAMES G ET AL) 4 Janvier 1994 * colonne 8, ligne 42 - colonne 9, ligne 50 * * colonne 11, ligne 28 - colonne 12, ligne 36; revendications; figure 3 * ---	1,2	G06F1/035 G06F7/52 G06F7/544
A	US-A-4 159 527 (YAHATA HARUKI ET AL) 26 Juin 1979 * colonne 8, ligne 15 - ligne 39; revendication 2; figure 8 * ---	1	
A	FR-A-2 208 554 (LABORATOIRE CENTRAL DE TÉLÉCOMMUNICATIONS) 21 Juin 1974 * le document en entier * ---	1-5	
A	GB-A-2 152 715 (CASIO COMPUTER CO LTD) 7 Août 1985 * abrégé; figures 2,6 * -----	1,5	
			DOMAINES TECHNIQUES RECHERCHES (Int.Cl.6)
			G06F

Le présent rapport a été établi pour toutes les revendications

Lieu de la recherche	Date d'achèvement de la recherche	Examinateur
BERLIN	15 Février 1996	Durand, J

CATEGORIE DES DOCUMENTS CITES

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : arrière-plan technologique
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant

EPO FORM 1503 03.82 (P04C02)